Implementation of QT Algorithm for BBC-Small Tiles and VPD
qt32b_l0_v5_3.mcs

Chris Perkins

03/02/2009

Description:

This algorithm forms a 16bit ADC Sum and 12bit TAC Max. Only channels that satisfy a "good hit" requirement are included in the ADC Sum and TAC Max. A "good hit" is defined as one where the ADC value is greater than some threshold and the corresponding TAC value is greater than TAC_MIN and less then TAC_MAX. The channel mask register can be used but note that ADC and TAC channels must each be masked individually.

Inputs:

QT8A: 4 PMT ADC, 4 PMT TAC
QT8B: 4 PMT ADC, 4 PMT TAC
QT8C: 4 PMT ADC, 4 PMT TAC
QT8D: 4 PMT ADC, 4 PMT TAC

Registers (1 Set Per Daughter Card)**:**

Alg. Reg. 0 (Reg 13): ADC_Threshold
Alg. Reg. 1 (Reg 14): TAC_MIN
Alg. Reg. 2 (Reg 15): TAC_MAX
Reg. 11: Channel Mask

LUT:

Timing adjustments/pedestal subtraction for each PMT

Action (21x RHIC Clock)**:**

1st: Mask channels and Latch inputs
If mask bit = 1, channel data = 0

2nd: For each PMT (4 per daughter board):
ADC above threshold: ADC > PMT_ADC_Thresh → Good_ADC
TAC above threshold: TAC > TAC_MIN → Good_TAC_MIN
TAC below threshold: TAC < TAC_MAX → Good_TAC_MAX

3rd: Make good_hits(0-3):
good_hit(i) = Good-ADC(i) && Good_TAC_MIN(i) && Good_TAC_MAX(i)

4th: Sum channels 0+1 subject to good hit requirements → Int_sum_0
Sum channels 2+3 subject to good hit requirements → Int_sum_1
Compare TAC channels 4, 5 subject to good hit requirements → Int_max_0
Compare TAC channels 6, 7 subject to good hit requirements → Int_max_1

5th: Sum Int_sum_0 + Int_sum_1 → Int_sum_2
Compare Int_max_0, Int_max_1 → Int_max_2

6th: Sum Int_sum_2 + Sum from previous daughters → ADC_Sum
Compare Int_max_2 to TAC Max from previous daughters → TAC_Max

7th: Latch Output Bits to next daughter or L0 FPGA
(0-15) : ADC_Sum
(16) : '0'
(17-28) : TAC_Max
(29-33) : '0'

Algorithm Latch: 1 or 2

L0 Output to DSM:
(0-15) : ADC Sum
(16-27) : TAC Max
(28-31) : '0'